

APPENDIX A

Ink Object

The following interface may be used to return basic information about the ink object. It may also return information that is shared among strokes and points for this ink object. For example, if the points in this ink object are “cursor down” points, this will be reported through this interface. If the points are recorded using the same tablet, and that tablet supports pressure, then it is possible to find out that pressure is supported for all points.

This interface supports the following features:

10

Methods in Vtable Order

IUnknown Methods	Description
QueryInterface	Returns pointers to supported interfaces.
AddRef	Increments reference count.
Release	Decrements reference count.

Methods	Description
GetAttribute	This function gets ink object attributes to a provided buffer location.
GetBoundingBox	This function returns the bounding rectangle of the ink in the ink space coordinates. The rectangle coordinates are in same units as the ink coordinates. The bounding box will take into account the pen width of the strokes in the ink object as well as other drawing attributes that apply for each stroke. For example, if a given stroke is drawn as a Bezier fitted curve with variable pressure, the method will correctly take into account all of these facts when computing the bounding box.
GetPropertyList	This function returns to a provided buffer location the array of GUIDs of the properties that are global for this Ink Object.
GetPropertyData	This function is used by an application to query the ink object for the value associated with global ink property specified by GUID. If GUID passed in is not one of the global ink properties in the ink object the function fails.
SetPropertyData	This function is used by the application to create or change global ink property specified by GUID. Calling SetPropertyData for a GUID not in the Ink Object results in the property being created. If the property already exists then the value is overwritten. Predefined GUIDs values have an explicit type.
DeletePropertyData	This function deletes the ink property data of the property identified by GUID from the ink object.
GetDrawingAttributes	Returns a set of drawing attributes associated with a cid for this ink object.

	Calling <code>IInkObject::GetDrawingAttributes</code> before calling <code>IInkObject::SetDrawingAttributes</code> will return the default set of drawing attributes. The drawing attributes associated with a given stroke can be obtained by calling <code>IInkStroke::GetDrawingAttributes</code> .
<code>SetDrawingAttributes</code>	This function sets the drawing attributes for all the future strokes that will be drawn with this cursor id. If the ink is being generated by the application and is not coming from a TC then set CID of zero. If no drawing attributes are explicitly set for a cursor then the default drawing attributes are assumed. Calling <code>IInkObject::SetDrawingAttributes</code> does not increase the reference count on the Drawing attributes object since the attribute values are copied and saved in the Ink Object. Calling <code>IInkObject::SetDrawingAttributes</code> for the second time with different attributes but the same cursor id will only cause subsequent strokes to be drawn with the new attributes but does not affect strokes already drawn. The drawing attributes can later be changed by calling <code>IInkStroke::ModifyDrawingAttributes</code> or <code>IInkObject::ModifyDrawingAttributes</code> .
<code>IInkObject::ModifyDrawingAttributes</code>	This function modifies the drawing attributes of the already existing strokes, regardless of the cursor id that was originally used to record the strokes. Calling <code>SetDrawingAttributes</code> does not increase the reference count on the Drawing attributes object since the attribute values are copied and saved in the Ink Object. Calling <code>IInkObject::ModifyDrawingAttributes</code> only affects the strokes specified by a stroke set and has no effect on the strokes that will be written in the future. The drawing attributes that will be drawn in the future will use the set of drawing attributes specified by <code>IInkObject::SetDrawingAttributes</code> .
<code>GetInkSpaceRectangle</code>	This function returns an Ink Space Rectangle for the current ink object.
<code>SetInkSpaceRectangle</code>	This function sets the Ink Space Rectangle for the current ink object.
<code>GetStrokeCount</code>	Returns a count of the strokes in the Ink Object.
<code>GetStroke</code>	Returns a stroke in the Ink Object.
<code>MergeStrokes</code>	Merges a contiguous range of strokes into a single stroke.
<code>DeleteStrokes</code>	Deletes Strokes from a specified starting point.
<code>CreateStrokeSet</code>	Creates empty stroke set
<code>DeleteStrokeSet</code>	Deletes a stroke set

ExtractStrokeSet	This function creates a new ink object from a subset of strokes in the original ink object. Subset of strokes is specified as either strokes not contained in an identified stroke set or those contained in this stroke set, depending on a flag. The strokes in the original ink object are optionally deleted, depending on whether a flag is set.
Clone	Creates a new ink object, in memory, that is an exact clone of the original. The reference count of the new ink object is set to one upon successful return from this function.
Extract	Delimits ink to be “cut” (as in cut and paste operation) or extracted out of an existing ink object.
Trim	This function moves the bounding box of the ink to the origin and resets the Ink Space Rectangle of the ink object to be equal to its bounding box that is now aligned with the origin.
IInkObject::Clip	This function clips the whole ink object to a specified rectangle.
MergeSimple	Merges (or “pastes”) the Source Ink Object into the current Ink Object at the target point. The strokes of the Source Ink Object are sequentially appended after the strokes of the existing ink object and the coordinates are modified so that the ink is merged starting at the specified point.
Merge	Merges (or “pastes”) the Source Ink Object into the current Ink Object and scales it into the specified rectangle. The strokes of the Source Ink Object are sequentially appended after the strokes of the existing ink object and the coordinates are modified so that the ink is merged and scaled inside the rectangle.
HitTest	Use this function to find out which strokes, if any, when drawn taking into account the full set of drawing attributes, intersect a given area.
SetCompressionMode	This function sets the compression mode used when this object is serialized using IPersistStream::Save function.

IStrokeCollect

The IStrokeCollect interface may be used to collect packets into a stroke. Supports the notion of adding ink to several strokes at the same time.

5

This interface supports the following features:

Methods in Vtable Order

Unknown Methods	Description
QueryInterface	Returns pointers to supported interfaces.
AddRef	Increments reference count.
Release	Decrements reference count.

Methods	Description
BeginStroke	This function is called to start the collection of a new stroke. Typically this function is called when the pen enters the proximity of the tablet or when it touches the tablet surface. Two or more strokes can be collected at the same time, provided that they correspond to distinct cursor identifiers. The stroke index is valid until EndStroke is called for the CID.
AppendPackets	Add the specified packet to the stroke being constructed.
EndStroke	Call this function when all packets have been added to the stroke (usually called on a cursor up).

10

IIPersistStream

Derived from standard COM IPersist interface. It has four methods:

5 Methods in Vtable Order

Unknown Methods	Description
QueryInterface	Returns pointers to supported interfaces.
AddRef	Increments reference count.
Release	Decrements reference count.

Methods	Description
IPersistStream::IsDirty	Tells whether object changed since it was last saved.
IPersistStream::Load	Loads the object
IPersistStream::Save	Saves the object and optionally clears the dirty flag.
IPersistStream::GetSizeMax	Compute the size of stream needed to store the whole object in the stream. That is the size after any compression.

ITransformInk

This interface is used to perform simple geometric transformations on the ink object.

5 This interface supports the following features:

Methods in Vtable Order

Unknown Methods	Description
QueryInterface	Returns pointers to supported interfaces.
AddRef	Increments reference count.
Release	Decrements reference count.

Methods	Description
ITransformInk::TranslateInk	Translate a subset of strokes of an ink object by a specified x and y amount.
ITransformInk::ScaleInk	Scale a subset of strokes of an ink object by a specified amount.
ITransformInk::RotateInk	Rotate a subset of strokes of an ink object by a specified amount.
ITransformInk::TransformInk	Apply a linear transform to a subset of strokes of an ink object.
ITransformInk::FitInkToRectangle	Scale and translate ink coordinates so that ink's bounding box is as specified.

IRenderInk

This is the interface to Ink Rendering Services. It includes functions for drawing and measuring the ink output. The measuring functions closely parallel drawing functions.
5 This interface supports the following features:

Methods in Vtable Order

Unknown Methods	Description
QueryInterface	Returns pointers to supported interfaces.
AddRef	Increments reference count.
Release	Decrements reference count.

Methods	Description
IRenderInk::DrawInkSimple	Function used to draw the ink object. The ink is drawn in its natural aspect ratio, i.e. as it is drawn on the tablet, only it is scaled to fit the specified height on the display and placed relative to the origin parameter as specified by an alignment parameter. The origin is placed along the sides of the resulting bounding box of the ink as specified by the ink alignment values in the alignment parameter.
IRenderInk::DrawInk	Transforms the coordinates in the ink object to the GDI logical coordinates that all GDI drawing routines take as an input
IRenderInk::DrawInkInRectangle	Draws a selected rectangular area in the ink space in a prescribed rectangular area in the GDI logical space.
IRenderInk::MeasureInkSimple	Returns the logical ink coordinates of the bounding box in the application provided buffer. Returns the width given the height of the ink, and takes into account the width of the pen. When ink height is specified in absolute units (e.g., in points), the result is returned in hdc logical units. In this case, hdc needs to be passed to a function for conversion between physical units to logical hdc units.
IRenderInk::MeasureInk	This function returns the bounding box of an ink when ink is drawn with a certain transform. Takes into account drawing attributes that apply to the strokes, such as the pen width, curve fitting etc. Pressure and angles may be taken into account and when ink is drawn with curve fitting.

ITrimInk

5 This interface is used to filter out the information from the ink object that is not needed or desired by a given application. For instance, removing collinear points will result in the same drawing output, assuming pressure and other packet properties are not taken into account.

This interface supports the following features:

Methods in Vtable Order

Unknown Methods	Description
QueryInterface	Returns pointers to supported interfaces.
AddRef	Increments reference count.
Release	Decrements reference count.

Methods	Description
ITrimInk::RemoveCollinearPoints	This function removes all packet data associated with collinear points.
ITrimInk::DeleteEraserStrokes	Removes eraser data.
ITrimInk::DeletePenUpStrokes	Removes pen up data.
ITrimInk::DeletePacketPropertyData	This function removes data for the specified packet property from all ink packets in this ink object.